Model Checking Higher-Order Computation

C.-H. Luke Ong
Oxford University, United Kingdom

Over the past decade, model checking and allied methods have made great strides in the verification of first-order imperative programs, producing impressive tools that can readily verify properties of hundreds of thousands of lines of code. A challenge facing software verification research is to extend these methods and techniques to the formal analysis of higher-order programs. This lecture course presents recent progress in the application of semantic methods to the verification of higher-order computation, covering mainly foundational work but also offering a glimpse of experimental tool construction.

Game semantics has emerged as a powerful method to construct highly accurate models of programming languages that lend themselves to algorithmic analysis. As a case study, we consider the fully abstract game semantics of the 3rd-order fragment of Idealized Algol (a higher-order procedural language), which is representable as visibly pushdown automata (VPA). Thanks to game semantics, we show that observational equivalence is decidable by reduction to the VPA equivalence problem. [4, 2]

Models of computation and their algorithmics. There has been a recent revival of interests in higher-order recursion schemes as generators of infinite structures such as infinite trees and graphs. As a definitional device, they are highly expressive and robust: they are equi-expressive with a new class of higher-order collapsible pushdown automata. The structures they generate enjoy strong model-checking properties. We present a new type-theoretic proof that the modal μ-calculus model checking of these trees are n-EXPTIME complete. [5, 1, 3]

Flow analysis and reachability. We consider reachability in higher-order programs and its connections with ßow analysis. We formulate reachability as a decision problem in PCF, and show that even in the recursion-free fragment generated from a finite base type, several versions of the reachability problem are undecidable from order 4 onwards, several other versions are reducible to each other; and identify a decidable case. We characterise a version of the reachability problem in terms of a new class of tree automata introduced by Stirling called alternating dependency tree automata and examine some consequences. [6]

References

1. M. Hague, A. S. Murawski, C.-H. L. Ong, O. Serre. *Collapsible Pushdown Automata and Recursion Schemes*. Proc. "23rd IEEE Symposium on Logic in Computer Science" (LICSÕ08). IEEE Computer Society, 2008.
2. D. Hopkins, C.-H. L. Ong. *Homer: A Higher-order Observational-equivalence Model CheckER*. Proc. "Computer-Aided Verification" (CAVÕ09), LNCS, Springer-Verlag, 2009.
3. N. Kobayashi, C.-H. L. Ong. *A Type Theory Equivalent to the Modal μ-Calculus Model Checking of Higher-Order Recursion Schemes*. Proc. "24th IEEE Symposium on Logic in Computer Science" (LICSÕ09), IEEE Computer Society, 2009.
4. A. Murawski, I. Walukiewicz. *Third-Order Idealized Algol with Iteration is Decidable*. Proc. "FOS- SACS", LNCS 3441, pp. 202–218, Springer-Verlag, 2005.
5. C.-H. L. Ong. *On Model-Checking Trees Generated by Higher-Order Recursion Schemes*. Proc. "21st Annual IEEE Symposium on Logic in Computer Science" (LICSÕ06), pp. 81–90, Computer Society Press, 2006. Long version (55 pp.) available at `users.comlab.ox.ac.uk/luke.ong/` .
6. C.-H. L. Ong, T. Tzevelekos. *Functional Reachability*. Proc. "24th IEEE Symposium on Logic in Computer Science" (LICSÕ09), IEEE Computer Society, 2009.